

Analisis dan Implementasi Algoritma *Differential Evolution* Pada Jaringan Syaraf Tiruan Studi Kasus Prediksi Masa Studi Mahasiswa

Fahrudin Julianto¹, Jondri MSi², Rita Rismala ST.,MT³

Fakultas Informatika Telkom University
Jln. Telekomunikasi No.1 Kabupaten Bandung

¹fahrudin.julianto@gmail.com

²jondri@telkomuniversity.ac.id

³rrs@telkomuniversity.ac.id

Abstrak

Lama masa studi mahasiswa merupakan salah satu parameter penting dalam evaluasi performansi studi mahasiswa. Untuk itu, sangatlah wajar jika prediksi lama masa studi mahasiswa dibutuhkan oleh manajemen perguruan tinggi.

Pada penelitian ini dibangun sistem prediksi menggunakan Jaringan Syaraf Tiruan dengan *Differential Evolution* sebagai algoritma pelatihnannya. *Differential Evolution* merupakan salah satu dari *Evolution Algorithms*, algoritma-algoritma optimasi yang berbasis evolusi biologi yang ada di dunia nyata. Berbeda dengan metode *Evolution Algorithms* lainnya yang menggunakan mutasi acak, pada *Differential Evolution*, mutasi yang dilakukan adalah mutasi semi terarah. Hal ini sangat mempengaruhi pencarian solusi yang terbukti lebih cepat dibandingkan *Genetik Algorithm* dan *Evolution Strategies*. *Differential Evolution* juga cepat konvergen dan sangat mudah digunakan.

Data yang digunakan pada penelitian ini adalah data akademik yang berupa data Kartu Hasil Studi mahasiswa. Variabel yang digunakan sebagai input untuk sistem yang dibangun adalah nilai beberapa mata kuliah tahun pertama dan tahun kedua beserta berapa kali setiap mahasiswa mengambil mata kuliah tersebut dengan *output* sistem prediksi adalah mahasiswa lulus tepat waktu dan mahasiswa lulus tidak tepat waktu. Akurasi terbaik yang dihasilkan dari sistem yang dibangun adalah 75,9726.

Kata Kunci : *Differential Evolution*, Jaringan Syaraf Tiruan, prediksi masa studi mahasiswa.

Abstract

Study period is one of the important parameters in the evaluation of student performance. Therefore, it is natural that study period is required by university management.

In this research, prediction system is built by using Artificial Neural Networks with Differential Evolution as it's training algorithm. Differential Evolution is one of the Evolution Algorithms, optimization algorithm based on evolutionary biology that exist in the real world. In contrast to others Evolution Algorithms that use random mutation, at Differential Evolution, mutation is carried out semi-directed mutation that can greatly affect the search for solutions that proved faster than Genetic Algorithm and Evolution Strategies. Differential Evolution is fast convergence and easy to use.

The data used in this final project is academic data in the form of Kartu Hasil Studi. Variables are used as inputs to the prediction system is the value of several courses in the first and second years along with the number of times each student took the course. The output of prediction system is the students who graduate on time and the students who do not graduate on time. The best accuracy produced by sistem is 75,9726.

Keywords: *Artificial Neural Networks, Differential Evolution, study period.*

1. Pendahuluan

Semakin ketatnya persaingan dalam mendapatkan lapangan pekerjaan menuntut perguruan tinggi menghasilkan sarjana yang berkualitas dan berdaya

saing [3]. Oleh sebab itu, dalam manajemen pendidikannya, setiap perguruan tinggi selalu melakukan evaluasi performansi mahasiswa. Lama masa studi mahasiswa merupakan salah satu parameter penting dalam evaluasi performansi studi

mahasiswa. Mampu memprediksi masa studi mahasiswa akan memberikan manfaat yang besar bagi perguruan tinggi. Selanjutnya, prediksi masa studi mahasiswa dapat digunakan sebagai acuan dalam mengevaluasi proses pendidikan, kurikulum pendidikan dan kebijakan-kebijakan lain yang terkait performansi mahasiswa secara keseluruhan [2].

Beberapa penelitian terkait prediksi masa studi mahasiswa diantaranya adalah Prediksi Potensi Drop Out Mahasiswa dengan sistem yang dibangun dengan *Evolving fuzzy* menggunakan variabel prediktor IPK dan TPA (Tes Potensi Akademik) [1]. Penelitian lainnya adalah Prediksi Kelulusan Tepat Waktu yang dibangun dengan NEFCLASS menggunakan prediktor IPK dan beberapa variabel hasil analisis terhadap data mata kuliah tahun pertama (TPB) [2].

Mengacu pada penelitian-penelitian tersebut dan masih terus berkembangnya penelitian tentang prediksi masa studi mahasiswa, pada penelitian ini penulis membangun sistem prediksi masa studi mahasiswa dengan output berupa lulus tepat waktu dan lulus tidak tepat waktu. Variabel input yang digunakan adalah nilai beberapa mata kuliah tahun pertama dan tahun kedua beserta berapa kali setiap mahasiswa mengambil mata kuliah tersebut. Model prediksi yang digunakan adalah Jaringan Syaraf Tiruan (JST) dengan algoritma pelatihan adalah *Differential Evolution* (DE). DE merupakan salah satu algoritma optimasi yang berbasis evolusi biologi yang ada di dunia nyata.[8] Berbeda dengan metode EAs lainnya yang menggunakan mutasi acak. Pada DE, mutasi yang dilakukan adalah mutasi semi terarah dengan menggunakan *differential mutation* [8]. DE bisa menemukan minimum global dari fungsi multidimensional dan multimodal dengan probabilitas yang tinggi tanpa terpengaruh oleh nilai parameter awal.[8] DE baik digunakan untuk masalah optimasi bernilai *real* dan cepat konvergen.[6][8]

2. Dasar Teori

2.1 Jaringan Syaraf Tiruan

Jaringan syaraf tiruan (JST) adalah sistem pengolah informasi yang meniru sistem syaraf manusia[7]. Sama seperti sistem syaraf manusia yang tersusun oleh *neuron*, JST juga tersusun oleh *perceptron* (*neuron* pada JST) sebagai unit pemrosesan informasi yang merupakan dasar dari operasi JST [7].

2.1.1 Model Neuron

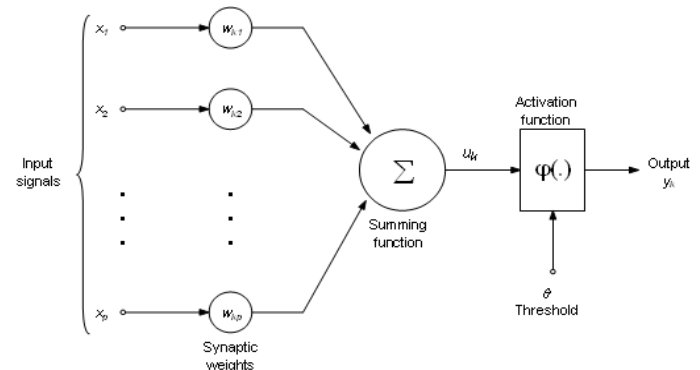
Terdapat tiga elemen dasar dari model *neuron*, yaitu[7] :

- Sekumpulan sinapsis yaitu jalur penghubung antar *neuron*, dimana masing-masing

sinapsis memiliki kekuatan hubungan atau bobot.

- Suatu *adder* untuk menjumlahkan sinyal-sinyal *input* yang sudah dikalikan dengan bobot sinapsis yang sesuai.
- Suatu fungsi aktivasi untuk menentukan *output* berdasarkan dari hasil penjumlahan sinyal-sinyal *input* dengan bobot sinapsis yang sesuai.

Contoh model neuron[7] :



Gambar 2.1 model neuron

2.1.2 Fungsi Aktivasi

Fungsi aktivasi digunakan untuk menentukan *output* dari suatu *neuron*[9].

Beberapa fungsi aktivasi yang sering digunakan pada JST, antara lain :

- Fungsi *Hard Limit*
- Fungsi *Symmetric Hard Limit*
- Fungsi *Linear*
- Fungsi *Sigmoid Biner*
- Fungsi *Sigmoid Bipolar*

2.1.3 Arsitektur Jaringan

Beberapa arsitektur jaringan yang sering dipakai dalam JST antara lain[7] :

- Single-Layer Network*
Sekumpulan unit/*neuron* pada *input layer* dihubungkan dengan sekumpulan unit pada *output layer*. Setiap unit *input* diproyeksikan dengan semua unit *output*.
- Multi-Layer Network*
Jaringan dengan satu atau lebih *hidden layer*. Setiap unit *input* diproyeksikan ke semua *hidden unit* pada *hidden layer* dan setiap *hidden unit* diproyeksikan ke setiap unit *output*. Jika *hidden layer* lebih dari satu, maka setiap unit *input* diproyeksikan ke semua *hidden unit* pada *hidden layer* pertama, kemudian setiap *hidden unit* pada *hidden layer* pertama diproyeksikan ke semua *hidden unit* pada *hidden layer* berikutnya dan seterusnya sampai *hidden layer* terakhir dan kemudian

dilanjutkan dengan proyeksi dari *hidden unit* pada *hidden layer* terakhir ke semua *unit output*.

c. *Recurrent Network*

Jaringan yang mempunyai minimal satu *feedback loop*. *Recurrent network* bisa terdiri dari satu atau lebih layer. *Unit output* pada jaringan ini memberikan kembali *outputnya* sebagai sinyal *input* pada semua *unit input*.

2.1.4 Proses Pelatihan

Proses pelatihan pada JST dapat didefinisikan sebagai suatu proses dimana parameter-parameter bebas JST diadaptasikan melalui suatu proses perangsangan berkelanjutan oleh lingkungan dimana jaringan berada [7].

Jenis pelatihan pada JST yaitu [7] :

- a. *Supervised Learning* (Pelatihan dengan pengawasan)
 Dalam poses pelatihan ini, JST diberikan suatu rangsangan dari lingkungan dimana pengetahuan tentang lingkungan tersebut sudah diketahui (biasa direpresentasikan dengan *input-output*). Dengan rangsangan tersebut JST akan memberikan respon. Respon tersebut merepresentasikan aksi yang dilakukan oleh JST. Parameter-parameter JST berubah-ubah berdasarkan rangsangan dan sinyal kesalahan (sinyal kesalahan adalah perbedaan antara *output* JST dan *output* yang diinginkan). Sinyal kesalahan digunakan sebagai umpan balik ke jaringan. Proses perubahan ini dilakukan terus-menerus sampai JST mampu memetakan sekumpulan *input-output* dengan akurasi yang tinggi.
- b. *Unsupervised Learning* (Pelatihan tanpa pengawasan)
 Dalam proses pelatihan ini, terdapat suatu *layer* pada jaringan dimana *neuron-neuronnya* bersaing meraih “kesempatan” untuk memberikan respon kepada jaringan tentang ciri khas dari data masukan. Berdasarkan dari ciri-ciri yang diberikan oleh *neuron* tersebutlah, jaringan menghasilkan *output*.

2.2 Differential Evolution

Differential Evolution (DE) merupakan suatu metode optimasi dengan pendekatan heuristik untuk mencari nilai minimum dari fungsi ruang kontinyu yang non-linier dan non-differentiable. DE merupakan salah satu dari *Evolution Algorithms*, algoritma-algoritma optimasi *meta-heuristic* yang berbasis mekanisme evolusi biologi yang ada di dunia nyata seperti mutasi, perkawinan atau rekombinasi, seleksi alam dll [8]. DE bisa menemukan minimum global dari

fungsi *multidimensional* dan *multimodal* dengan probabilitas yang tinggi [8]. DE mencari nilai minimum secara paralel menggunakan sejumlah individu dalam suatu populasi. Pada DE, individu-individu direpresentasikan dengan nilai-nilai *real* yang bisa dipandang sebagai suatu vektor. Individu baru didapatkan dengan menggunakan perhitungan tertentu berbasis pada perbedaan jarak vektor antar individu orang tua [8]. DE menggunakan sejumlah NP vektor parameter sebagai suatu populasi pada setiap generasi G.

$$\underline{X}_{i,G}, i = 0, 1, 2, \dots, NP-1 \quad (2.1)$$

Selama proses pencarian nilai minimum (minimasi), vektor parameter tetap berjumlah NP. Populasi awal dibangkitkan secara acak [8].

2.2.1 Representasi individu

DE menggunakan representasi bilangan real untuk merepresentasikan individu ke dalam bentuk kromosom dengan gen-gen yang bernilai real dimana individu bisa dipandang sebagai suatu vektor [5]. Perbedaan antara dua individu dapat dihitung sebagai jarak antara dua vektor [5].

2.2.2 Seleksi orang tua

Pemilihan orang tua dilakukan dengan probabilitas yang sama untuk setiap individu tanpa memperhatikan nilai fitness-nya [8]. Pemilihan orang tua dilakukan dengan menggunakan distribusi *uniform* [8].

2.2.3 Differential Mutation

Differential mutation merupakan proses untuk membangkitkan vektor individu baru dengan melibatkan individu orang tua [8]. Ada beberapa skema DE untuk membangkitkan vektor individu baru. Skema DE yang digunakan pada penelitian ini adalah Skema DE2. Skema DE2 melibatkan empat individu sebagai orang tua dimana tiga individu merupakan individu yang dipilih secara acak dan satu individu merupakan vektor terbaik pada generasi saat ini. Pada skema DE2, *differential mutation* dilakukan berdasarkan rumus :

$$\underline{V} = \underline{X}_{r1,G} + \lambda \cdot (\underline{X}_{best,G} - \underline{X}_{r1,G}) + F \cdot (\underline{X}_{r2,G} - \underline{X}_{r3,G}) \quad (2.2)$$

Dimana :

$\underline{X}_{best,G}$ adalah individu terbaik saat ini

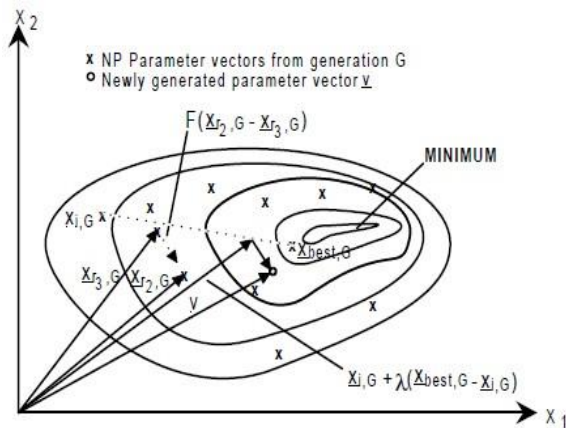
$r_1, r_2, r_3 \in [0, NP-1]$ dan r_1 adalah integer,

$F > 0$.

r_1, r_2, r_3 adalah bilangan integer, dimana satu dengan yang lain tidak boleh sama nilainya dan dipilih secara acak dalam interval $[0, NP-1]$. Ketiga bilangan tersebut menyatakan indeks orang tua dan bisa berbeda untuk setiap proses pembangkitan individu baru.

F adalah suatu bilangan real dan merupakan konstanta yang mengontrol penguatan *differential variation* ($\underline{x}_{r2,G} - \underline{x}_{r3,G}$). λ adalah suatu variabel kontrol bernilai real yang digunakan untuk mempertajam arah pencarian yang berhubungan dengan vektor terbaik saat ini[8].

Berikut adalah ilustrasi proses differential mutation skema DE2 :



Gambar 2.2 proses *differential mutation* skema DE2

2.2.4 Rekombinasi

Untuk meningkatkan keberagaman vektor-vektor parameter, maka vektor \underline{v} hasil *differential mutation* direkombinasikan dengan suatu vektor sembarang dalam populasi, misal vektor $\underline{x}_{i,G}$.

Proses rekombinasi menghasilkan vektor baru \underline{u} seperti berikut[8] :

$$\underline{u} = (\underline{u}_1, \underline{u}_2, \dots, \underline{u}_D)$$

rekombinasi dilakukan dengan cara :

$$\underline{u}_j = \underline{v}_j, \text{ untuk } j = \langle n \rangle_D, \langle n+1 \rangle_D, \dots, \langle n+L-1 \rangle_D$$

dan $\underline{u}_j = \text{untuk } j \text{ yang lainnya}$

dimana:

simbol $\langle \rangle_D$ menyatakan fungsi modulo dengan modulus D. Modulo adalah suatu fungsi dengan output sisa pembagian.

contoh : $\langle 8 \rangle_3 = 2$.

n merupakan bilangan bulat yang dipilih secara acak dalam interval $[0, D-1]$.

L merupakan bilangan bulat yang dipilih secara acak dalam interval $[0, D-1]$ dengan probabilitas $\Pr(L=v) = (CR)^v$ dimana $CR \in [0,1]$ adalah probabilitas *crossover*.

Pembangkitan bilangan acak untuk n dan L dilakukan untuk setiap pembangkitan vektor baru \underline{v} [8].

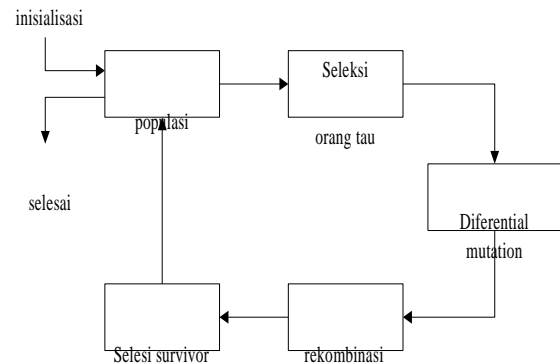
2.2.5 Seleksi survivor

Vektor hasil rekombinasi \underline{u} akan menggantikan vektor $\underline{x}_{i,G}$ pada generasi berikutnya jika \underline{u} menghasilkan nilai yang lebih baik daripada vektor $\underline{v}_{i,G}$. Tetapi jika \underline{u} memberikan nilai yang lebih buruk, maka \underline{u} tidak menggantikan vektor $\underline{x}_{i,G}$ yang berarti

vektor $\underline{x}_{i,G}$ akan tetap muncul pada generasi berikutnya[8].

2.2.6 Proses evolusi pada Differential Evolution

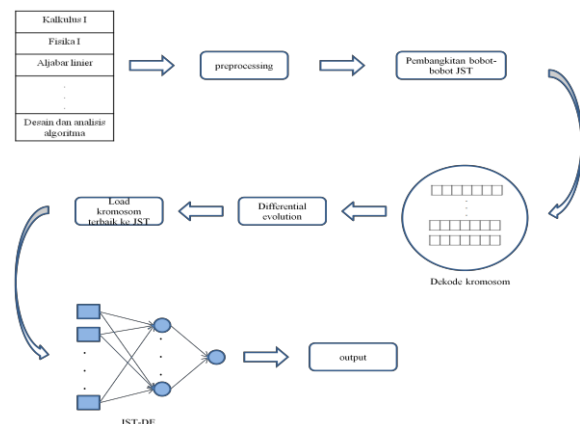
Berikut adalah proses evolusi pada DE[4] :



Gambar 2.3 proses evolusi pada

3. Perancangan Sistem

Pada penelitian ini dibangun sistem prediksi masa studi mahasiswa menggunakan Jaringan Syaraf Tiruan dengan algoritma pelatihan *Differential Evolution* untuk mengoptimasi bobot-bobot jaringan pada Jaringan Syaraf Tiruan. Berikut ini adalah alur sistem yang dibangun :



Gambar 3.1 Alur sistem prediksi masa studi mahasiswa.

Dilakukan *preprocessing* terhadap parameter-parameter yang digunakan sebagai input sistem. Selanjutnya, hasil *preprocessing* digunakan sebagai inputan JST untuk membangkitkan bobot-bobot jaringan. Pembangkitan bobot-bobot jaringan dilakukan secara berulang sejumlah populasi dari DE yang diinginkan. Setiap bobot-bobot hasil pembangkitan JST didekodekan menjadi sebuah kromosom. Selanjutnya kromosom-kromosom hasil decode digunakan untuk proses DE. Proses DE menghasilkan kromosom yang berisi bobot-bobot yang optimal. Selanjutnya, bobot-bobot yang optimal

hasil proses DE tersebut digunakan pada JST untuk diuji akurasi system yang dibangun.

4. Hasil Pengujian

Proses pengujian digunakan untuk mengetahui parameter-parameter apa saja yang mempengaruhi performansi sistem yang dibangun. Performansi sistem diukur berdasarkan akurasi data testing.

Berikut ini adalah scenario pengujian dan hasil pengujian dari sistem yang dibangun :

4.1 Skenario 1

parameter-parameter yang dibandingkan adalah :

- Persentase data training dan testing = 50-50 dan 80-20
- Jumlah neuron pada hidden layer = 5 dan 15
- Jumlah populasi DE = 50 dan 100
- Epoch DE = 500 dan 1000
- Dengan menggunakan parameter CR = 0,8, F = 0,8 dan lambda = 0,8

Tabel 4.1 hasil pengujian 1

Neuron pada hidden	Persentase data training	Populasi DE	Epoch DE	Max fitness DE	Akurasi data training	Akurasi data testing
5	50	50	500	87,9868	82,8804	70,2186
5	80	50	500	86,0531	80,2721	71,2329
5	50	50	1000	89,2916	85,3261	70,7650
5	80	50	1000	87,7917	83,5034	73,2877
5	50	100	500	87,4290	85,0543	68,8525
5	80	100	500	85,5943	74,6575	67,7596
5	50	100	1000	86,8443	79,6196	67,2131
5	80	100	1000	86,9777	81,2925	67,1233
15	50	50	500	84,8402	78,8043	71,0383
15	80	50	500	85,1617	80,6122	69,1781
15	50	50	1000	86,5201	80,1630	68,3060
15	80	50	1000	85,3629	79,2517	75,9726
15	50	100	500	85,8506	79,3478	71,5847
15	80	100	500	85,4465	79,5918	73,2877
15	50	100	1000	86,0	80,1	67,7

			0	270	630	596
15	80	100	100	84,9	77,7	69,1
			0	582	211	781

4.2 Skenario 2

parameter yang dibandingkan adalah parameter mutasi pada DE yaitu *Crossover Rate*, F, dan lambda. Untuk jumlah populasi, jumlah *epoch* pada DE, jumlah hidden neuron pada JST dan persentase data training dan data testing, akan menggunakan hasil dari pengujian skenario 1.

Adapun nilai dari parameter-parameter yang dibandingkan adalah :

- Nilai *Crossover Rate* yaitu 0,1, 0,5 dan 0,8
- Nilai F yaitu 0,1, 0,5 dan 0,8
- Nilai lambda yaitu 0,1, 0,5 dan 0,8

Tabel 4.2 hasil pengujian 2

Crossover Rate	F	Lambda	Max fitness	Akurasi data training	Akurasi data testing
0,1	0,1	0,1	85,813	79,591	69,178
0,1	0,1	0,5	86,261	80,442	69,178
0,1	0,1	0,8	86,380	80,442	71,917
0,1	0,5	0,1	86,628	80,272	69,178
0,1	0,5	0,5	87,001	81,972	71,917
0,1	0,5	0,8	87,193	81,462	73,972
0,1	0,8	0,1	85,636	79,061	71,917
0,1	0,8	0,5	86,654	81,972	71,917
0,1	0,8	0,8	86,912	81,632	73,287
0,5	0,1	0,1	82,513	73,129	71,232
0,5	0,1	0,5	81,167	69,387	69,178
0,5	0,1	0,8	82,039	70,918	71,917
0,5	0,5	0,1	87,119	80,442	69,863
0,5	0,5	0,5	87,061	82,142	70,547
0,5	0,5	0,8	87,280	81,802	72,602
0,5	0,8	0,1	83,736	77,381	67,123
0,5	0,8	0,5	85,247	79,591	60,274
0,5	0,8	0,8	86,318	79,081	67,808
0,8	0,1	0,1	81,770	69,727	69,178
0,8	0,1	0,5	80,309	69,287	69,178
0,8	0,1	0,8	80,990	69,387	69,178
0,8	0,5	0,1	89,592	86,904	67,808
0,8	0,5	0,5	85,488	78,571	69,863
0,8	0,5	0,8	84,344	77,721	72,602
0,8	0,8	0,1	83,666	77,210	72,602
0,8	0,8	0,5	86,654	81,972	71,232
0,8	0,8	0,8	85,362	79,251	75,972

4.3 Skenario 3

Pada skenario 3, dilakukan prediksi masa studi mahasiswa menggunakan Jaringan Syaraf Tiruan dengan algoritma pelatihan *BackPropagation*. Parameter-parameter pada Jaringan Syaraf Tiruan

dengan algoritma pelatihan *Back Propagation* sama seperti parameter pada Jaringan Syaraf Tiruan dengan algoritma pelatihan *Differential Evolution*. Parameter-parameter itu adalah :

- Persentase data training dan testing = 50-50 dan 80-20
- Jumlah neuron pada hidden layer = 5 dan 15
- Epoch* DE = 500 dan 1000
- Learning rate = 0,2, 0,5 dan 0,8

Tujuan dari skenario 3 adalah sebagai pembandingan untuk mengetahui performansi sistem prediksi yang dibangun menggunakan Jaringan Syaraf Tiruan dengan algoritma pelatihan *Differential Evolution* dilihat dari akurasi sistem prediksi yang dihasilkan oleh Jaringan Syaraf Tiruan dengan algoritma pelatihan *Back Propagation*.

Tabel 4.3 hasil pengujian skenario 3

Jumlah hidden	Data training	epoch	Learning Rate	akurasi data training	Akurasi data testing
5	50	500	0.2	73.369	72.677
5	50	500	0.5	70.380	70.765
5	50	500	0.8	71.467	70.765
5	50	1000	0.2	74.184	72.131
5	50	1000	0.5	71.467	70.491
5	50	1000	0.8	73.641	72.131
5	80	500	0.2	70.238	69.863
5	80	500	0.5	70.068	70.547
5	80	500	0.8	70.748	69.178
5	80	1000	0.2	69.727	69.863
5	80	1000	0.5	72.449	68.493
5	80	1000	0.8	72.108	67.808
15	50	500	0.2	73.097	72.950
15	50	500	0.5	75.543	72.131
15	50	500	0.8	72.282	70.765
15	50	1000	0.2	75.721	72.404
15	50	1000	0.5	74.184	69.125
15	50	1000	0.8	76.630	68.032
15	80	500	0.2	72.959	70.547
15	80	500	0.5	70.578	71.917
15	80	500	0.8	70.748	71.917
15	80	1000	0.2	73.299	71.917
15	80	1000	0.5	73.469	71.232
15	80	1000	0.8	71.258	70.547

5. Kesimpulan dan Saran

5.1 Kesimpulan

- Differential Evolution* bisa digunakan untuk proses pelatihan pada Jaringan Syaraf Tiruan. Berdasarkan hasil penelitian, akurasi yang dihasilkan Jaringan Syaraf Tiruan dengan menggunakan algoritma *Differential Evolution* berada dalam rentang nilai yang mirip dengan

akurasi yang dihasilkan oleh Jaringan Syaraf Tiruan dengan algoritma *Back Propagation* yang biasa digunakan untuk algoritma pelatihan Jaringan Syaraf Tiruan.

- Parameter F yang menyatakan konstanta yang mengontrol penguatan differential variation pada *Differential Evolution* dan lambda yang menyatakan variabel kontrol yang digunakan untuk mempertajam arah pencarian yang berhubungan dengan vektor terbaik saat ini adalah parameter yang mempunyai pengaruh yang dominan dalam proses pelatihan
- Akurasi prediksi terbaik yang dihasilkan dari pengujian adalah 75,9726.

5.2 Saran

- Menggunakan metode preprocessing tertentu untuk mengolah data mentah
- Menggunakan algoritma pelatihan yang lain untuk proses pelatihan pada JST
- Menggunakan data *training* dan data *testing* yang lebih banyak
- Memprediksi masa studi mahasiswa dengan *output* berupa tahun atau semester yang ditempuh mahasiswa.

Daftar Pustaka

- [1] Adha, Rahmadhil. 2011. Penggunaan Algoritma Genetika dan Sistem Fuzzy Untuk Prediksi Potensi Drop Out Mahasiswa IT Telkom 2011
- [2] Fatriandini, Serli. 2013. Pembangunan Model Prediksi Kelulusan Tepat Waktu menggunakan Neuro Fuzzy. Bandung. Universitas Telkom
- [3] Meinanda, dkk. 2009. Prediksi Masa Studi Sarjana dengan Artificial Neural Network. Bandung. Institut Teknologi Bandung
- [4] Rizki Syaherli Chan, Mohd. 2012. Analisis dan Implementasi Differential Evolution untuk optimasi Artificial Neural Networks pada Prediksi Konsumsi Bahan Bakar Minyak. Bandung. Universitas Telkom
- [5] Rismala, Rita. 2011. Analisis Perbandingan Evolution Strategies dan differential Evolution pada Prediksi Data Time Series Studi Kasus Kurs Jual Emas. Bandung. Institut Teknologi Telkom
- [6] Sttart, Abdul. 2008. Differential Evolution for Neural Networks Learning Enhancement. Malaysia. Universiti Teknologi Malaysia
- [7] Suyanto.ST., MSc 2011. Artificial Intelligence Searching, Reasoning, Planning dan Learning. Bandung. Informatika
- [8] Suyanto.ST., MSc 2008. Evolutionary Computation Komputasi Berbasis "Evolusi" dan "Genetika". Bandung. Informatika

- [9] Wisesty, Untari Novia. 2010. Analisis dan Implementasi Algoritma Conjugate Gradient Polak Ribiere Untuk Pelatihan Backpropagation Studi Kasus Sistem Peramalan Temperatur Udara. Bandung. Institut Teknologi Telkom
- [10] http://unair.ac.id/unair_v1/filer/evstudmhs.pdf
diakses pada tanggal 25 Maret 2014